



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií ■

# NÁVRH A TVORBA DATABÁZE DOPRAVNÍCH NEHOD V ČR

## Bakalářská práce

*Studijní program:* B2646 – Informační technologie  
*Studijní obor:* 1802R007 – Informační technologie  
*Autor práce:* **Luděk Veselý**  
*Vedoucí práce:* Ing. Pavel Tyl





TECHNICAL UNIVERSITY OF LIBEREC  
Faculty of Mechatronics, Informatics  
and Interdisciplinary Studies ■

# DATABASE DESIGN AND IMPLEMENTATION FOR ROAD ACCIDENTS IN THE CZECH REPUBLIC

**Bachelor thesis**

*Study programme:* B2646 – Information technology  
*Study branch:* 1802R007 – Information technology  
*Author:* **Luděk Veselý**  
*Supervisor:* Ing. Pavel Tyl



## **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Luděk Veselý**  
Osobní číslo: **M12000190**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Název tématu: **Návrh a tvorba databáze dopravních nehod v ČR**  
Zadávací katedra: **Ústav mechatroniky a technické informatiky**

### Z á s a d y   p r o   v y p r a c o v á n í :


1. Navrhněte databázový informační systém pro evidenci dopravních nehod za účelem pozdějšího nasazení do dataminingové studie.
2. Prozkoumejte veřejné zdroje relevantních dat.
3. Naprogramujte serverovou a klientskou část DBS, vývojové nástroje zvolte po poradě s vedoucí práce.
4. Naplňte DBS daty z veřejných zdrojů.
5. Navrhněte další možné využití dat DBS dataminingovými postupy.

Rozsah grafických prací: dle potřeby dokumentace  
Rozsah pracovní zprávy: 30–40 stran  
Forma zpracování bakalářské práce: tištěná/elektronická  
Seznam odborné literatury:


- [1] Hana Kanisová, Miroslav Müller: UML srozumitelně, Computer Press, a.s., 2006
- [2] Steve McConnell: Dokonalý kód, Computer Press, a.s., 2006
- [3] Forrest Houlette.: SQL - příručka programátora, SoftPress, 2000
- [4] Šešera L. a kol.: Datové modelování v příkladech, Grada
- [5] Knuth Donald E.: Umění programovat, Cpress, 2008
- [6] Olivia Parr Rud: Datamining, Computer Press, a.s., 2006
- [7] <http://www.msps.cz/data-mining/>

Vedoucí bakalářské práce: Ing. Pavel Tyl  
Ústav mechatroniky a technické informatiky

Datum zadání bakalářské práce: 10. října 2013  
Termín odevzdání bakalářské práce: 16. května 2014

  
prof. Ing. Václav Kopecký, CSc.  
děkan



  
doc. Ing. Milan Kolář, CSc.  
vedoucí ústavu

V Liberci dne 10. října 2013

## Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

## Abstrakt

Tato bakalářská práce popisuje návrh a tvorbu databáze dopravních nehod v ČR. Práce se nejprve zabývá dostupnými zdroji relevantních dat. Na základě těchto dat je navržen databázový systém, který je následně implementován v prostředí Oracle 11g. Poté je naplněn získanými daty a vytvořena klientská aplikace. V závěru práce popisuje možné využití dat v dataminingové studii.

## Klíčová slova

Oracle, SQL, data mining, dopravní nehody, BASH, Java, shluková analýza, K-Means

## Abstract

This bachelor thesis describes creation of database of road accidents in the Czech Republic. Firstly, this thesis writes about available sources of data. On the ground of these data is designed database system, which is implemented in Oracle 11g. Secondly is filled the database by downloaded data and designed an client application. Finally is suggested future utilization of data in datamining study.

## Key words

Oracle, SQL, data mining, road accidents, BASH, Java, cluster analysis, K-Means

## **Poděkování**

Rád bych poděkoval vedoucímu práce Ing. Pavlovi Tylovi za rady a pomoc při řešení.

# Obsah

Seznam zkratek . . . . .	8
<b>1 Úvod</b>	<b>9</b>
<b>2 Použité hardwarové nástroje</b>	<b>11</b>
2.1 Lokální PC pro analýzu dat . . . . .	11
2.2 Server pro stahování dat . . . . .	12
<b>3 Zdroje dat</b>	<b>13</b>
<b>4 Získání dat</b>	<b>14</b>
4.1 Stažení identifikátorů nehod . . . . .	14
4.2 Stahování detailů nehod . . . . .	17
4.3 Konverze stažených dat . . . . .	18
<b>5 Návrh databáze</b>	<b>20</b>
5.1 Struktura zdrojových dat . . . . .	20
5.2 Normalizace databázového systému . . . . .	22
5.3 Naplnění databáze daty . . . . .	23
<b>6 Klientská část databázového systému</b>	<b>26</b>
<b>7 Využití dat v dataminingové studii</b>	<b>28</b>
7.1 Nejčastější příčiny tragických nehod . . . . .	28
7.2 Řešení úlohy pomocí nástroje Oracle Dataminer . . . . .	29
7.3 Nehody způsobené vysokou rychlostí . . . . .	31
7.4 Nejrizikovější křižovatky . . . . .	32
7.5 Rizikovost řidičů pro pojišťovny . . . . .	32
<b>8 Závěr</b>	<b>34</b>
<b>A Obsah přiloženého DVD</b>	<b>38</b>
<b>B Relační model databázového systému</b>	<b>39</b>
<b>C Výsledek shlukové analýzy</b>	<b>40</b>

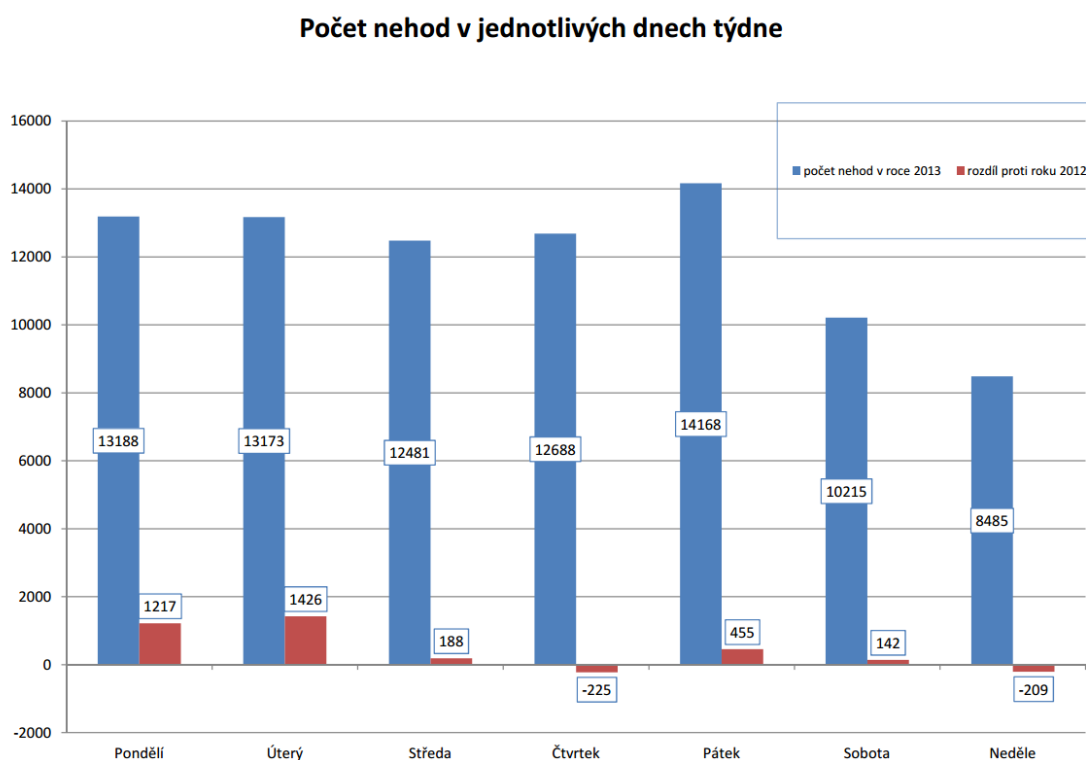


## Seznam zkratek

<b>OGI</b>	Open Government Licence
<b>DBS</b>	Databázový systém
<b>OS</b>	Operační systém
<b>VPS</b>	Virtuální privátní server, server běžící na virtualizovaném hardware
<b>SSH</b>	Secure Shell, zabezpečený komunikační protokol v počítačových sítích
<b>PHP</b>	PHP: Hypertext Preprocessor, skriptovací programovací jazyk
<b>BASH</b>	Bourne again shell, skriptovací jazyk
<b>CRON</b>	softwarový démon, který v operačním systému slouží jako plánovač úloh
<b>DDL</b>	Data definition language, jazyk definující databázové schéma
<b>SQL</b>	Structured Query Language, standardizovaný dotazovací jazyk pro práci s daty v relačních databázích
<b>PL/SQL</b>	Procedurální nadstavba jazyka SQL firmy Oracle
<b>CSV</b>	Comma separated values, formát pro výměnu tabulkových dat
<b>DM</b>	Data mining, analytická metodologie získávání netriviálních informací z dat
<b>KDD</b>	Knowledge discovery from databases, získávání znalostí z databází
<b>JDK</b>	Java Development Kit, sada nástrojů pro vývoj aplikací pro platformu Java
<b>JDBC</b>	Java Database Connectivity, rozhraní pro přístup k relačním databázím z programovacího jazyka Java

# 1 Úvod

V této bakalářské práci popisuji postup nalezení veřejně přístupných zdrojů dat o dopravních nehodách, stažení těchto dat, uložení dat ve vhodné databázi a možnost jejich využití v dataminingové studii. Ministerstvo dopravy poskytuje od roku 2006 v rámci projektu Jednotná dopravní vektorová mapa [1] databázi dopravních nehod na území ČR. Zde je možné nehody vyhledávat, zobrazit v mapě a ke každé nehodě získat podrobný výpis. Data by však bylo možné využít efektivněji. Policie ČR sice pravidelně vytváří statistiky nehodovosti [2], jde však pouze o jednoduché promítnutí jednotlivých parametrů nehod do grafu. Zpravidla se jedná o zobrazení hodnot zkoumané veličiny v daném období a porovnání s předchozím obdobím:



Obrázek 1.1: Počty nehod v jednotlivých dnech dle Policie ČR

Pokud by šlo pouze o procvičení dataminingových postupů nad databází dopravních nehod, bylo by možné použít již připravené volně dostupné soubory dat vytvořené přímo pro tento účel. Například Velká Británie taková data zveřejňuje pod licencí OGL [3] na svých webových stránkách [4]. V České Republice sice vznikl

projekt opendata.cz [5], projekt je ale v ranném stádiu a dostupných dat je velmi omezené množství. Dostupné jsou veřejné zakázky, hospodaření obcí, rejstříky škol a výsledky voleb 2006 a 2010.

The screenshot shows a web application titled "Dopravní nehody" (Traffic Accidents) with the URL "pcr.jdvm.cz/pcr/". The application has a search form with the following fields:

- Číslo nehody: (text input)
- Druh nehody: (dropdown menu, currently showing "srážka s chodcem")
- Alkohol: (dropdown menu, currently showing "ano, obsah alkoholu v krvi do 0,99‰")
- Viditelnost: (dropdown menu)
- Druh vozidla: (dropdown menu, currently showing "autobus")
- Počet vozidel: (dropdown menu, currently showing "je rovno") and a text input field.
- Následek nehody: nehody s následkem na zdraví osob (checkbox)
- umrceno osob: (dropdown menu, currently showing "je rovno") and a text input field.
- těžce zraněno: (dropdown menu, currently showing "je rovno") and a text input field.
- lehce zraněno: (dropdown menu, currently showing "je rovno") and a text input field.
- Zavinění nehody: (dropdown menu)
- Únik hmot: (dropdown menu)
- Číslo silnice: (text input)
- Obec: (text input) and a "Vyhledat obce" button.
- Datum od: (calendar icon) 01.01.2007 and Datum do: (calendar icon) 31.03.2014.
- A "Vyhledat" button and an "X" button.

Below the search form, there is a section titled "Nalezené nehody - celkem 1:" (Found accidents - total 1:). It shows a list of results with the header "(číslo nehody - datum):". The first result is "002100070013 - 01.01.2007". To the right of the list, there are two links: "Základní informativní výpis o nehodě (PDF)" and "Zobrazení v mapě". At the bottom left of the results section, there is a checkbox labeled "vybrat vše" (select all).

Obrázek 1.2: Aplikace PČR pro vyhledávání dopravních nehod

Cílem práce je tedy data o dopravních nehodách dostupná na serveru ministerstva dopravy převést do takové podoby, kdy bude možné jejich zpracování a dále navrhnout způsob tohoto zpracování. V první řadě půjde o vytvoření textového, strojově čitelného souboru, v druhé pak o uložení dat do vhodně navrženého DBS.

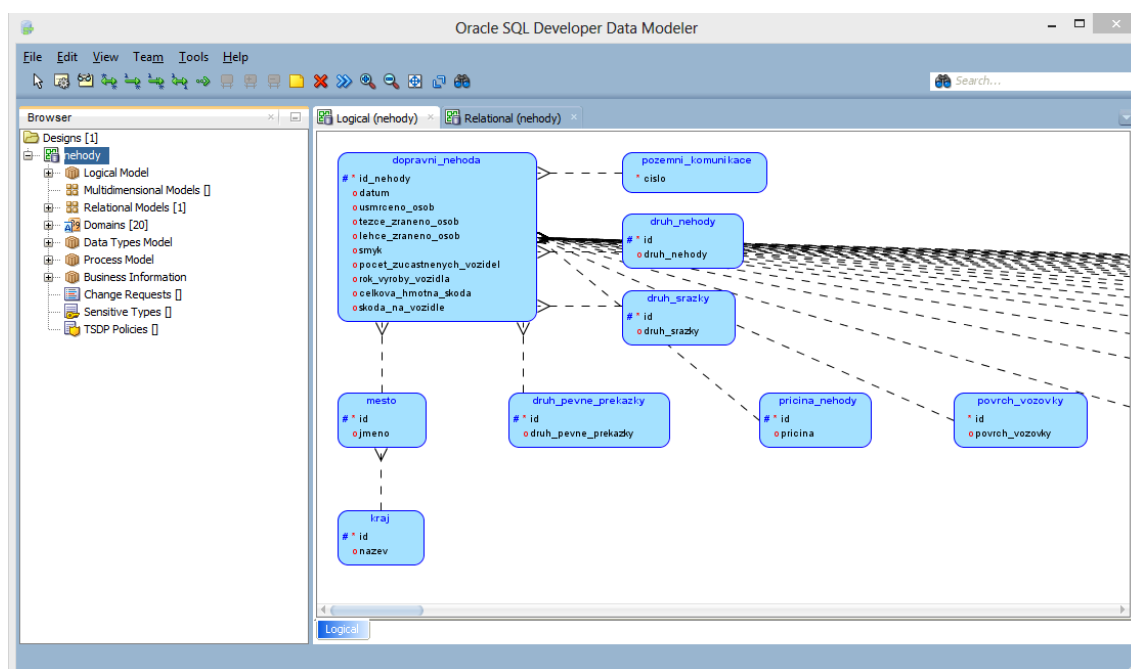
Jednotlivé body zadání zpracuji v posměněném pořadí. Nejprve prozkoumám zdroje relevantních dat, následně data postahuji. Na jejich základě navrhu a vytvořím databázovou aplikaci, kterou poté naplním daty. Na závěr navrhu využití těchto dat v dataminingové studii.

## 2 Použité hardwarové nástroje

### 2.1 Lokální PC pro analýzu dat

K dispozici jsem měl běžný počítač s OS Windows 8, který jsem využil k uložení dat do databáze a následné dataminingové studii. Také jsem zde provozoval nástroje pro návrh a běh DBS.

Pro uložení dat jsem se rozhodl využít databázi Oracle 11g [6]. Její použití je pro tento projekt vhodné z několika důvodů. Oracle nabízí řadu nástrojů, které s databází 11g spolupracují, vystačím si tedy s jediným úložištěm, další výhodou je její výkonnost a škálovatelnost. Pro využití v dataminingu se nabízí i další komerční nástroje, například IBM SPSS Modeler [7], SAS Enterprise Miner [8] nebo SAP BI [9]. Z nekomerčních nástrojů je nejrozšířenější programovací jazyk R [10].



Obrázek 2.1: Oracle SQL Developer Data Modeler

Nástrojů které lze použít pro analýzu dat je celá řada a některé jsou dostupné zcela zdarma, zmíněné nástroje však patří mezi nejkomplexnější. Pro získání podrobnějšího přehledu doporučuji prozkoumat [11, strana 25].

Pro návrh struktury databáze jsem využil nástroj Oracle SQL Developer Data Modeler [12]. Vytvořený model lze přímo transformovat do skriptu DDL a rovnou tak vytvořit databázové schéma. Navíc nástroj kontroluje, zda je model validní vzhledem k možnostem Oracle 11g, například délku názvu tabulky nebo použité znaky.

Dále jsem využil nástroj Oracle SQL Developer [13], který umožňuje správu samotné databáze, import dat, jejich zobrazení a manipulaci s nimi. Přímo v SQL Developeru je možné využít rozšíření Oracle Data Miner, ve kterém lze provést samotnou analýzu dat. V omezené míře nástroj umožňuje také pracovat s modely vytvořenými v Data Modeleru.

ID	MESTO	KRAJ	DATUM	ROK	MESIC	DEN_V_MESICI	DEN_V_TYDNU	CAS
481	002100070391	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 08:59
482	002100070392	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 09:10
483	002100070393	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 10:30
484	002100070394	Praha	Hlavní město Praha	09.01.2007	2007	01	09	TUESDAY 00:00
485	002100070395	Praha	Hlavní město Praha	09.01.2007	2007	01	09	TUESDAY 00:00
486	002100070396	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 10:05
487	002100070397	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 10:45
488	002100070399	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 12:20
489	002100070400	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 00:00
490	002100070401	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 12:40
491	002100070402	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 12:55
492	002100070403	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 12:55
493	002100070405	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 09:30
494	002100070406	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 13:30
495	002100070407	Praha	Hlavní město Praha	09.01.2007	2007	01	09	TUESDAY 00:00
496	002100070408	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 10:45
497	002100070409	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 13:25
498	002100070410	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 13:00
499	002100070411	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 13:40
500	002100070412	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 14:30
501	002100070413	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 14:30
502	002100070414	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 15:00
503	002100070415	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 15:10
504	002100070416	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 15:05
505	002100070417	Praha	Hlavní město Praha	10.01.2007	2007	01	10	WEDNESDAY 15:25

Obrázek 2.2: Oracle SQL Developer

Při instalaci a konfiguraci databáze jsem postupoval podle návodu a doporučení v knize Mistrovství v Oracle Database 11g [14].

## 2.2 Server pro stahování dat

Server jsem použil pro automatické stahování dat. Konkrétně šlo o VPS s distribucí Gentoo verze 2.2. Připojení k němu bylo možné pomocí SSH. K dispozici však nebyl root přístup, bylo tedy třeba si vystačit s výchozí instalací. Na tomto serveru probíhalo stahování a průběžná konverze postahovaných dat do podoby, kterou už bylo možné zpracovat na lokálním PC.

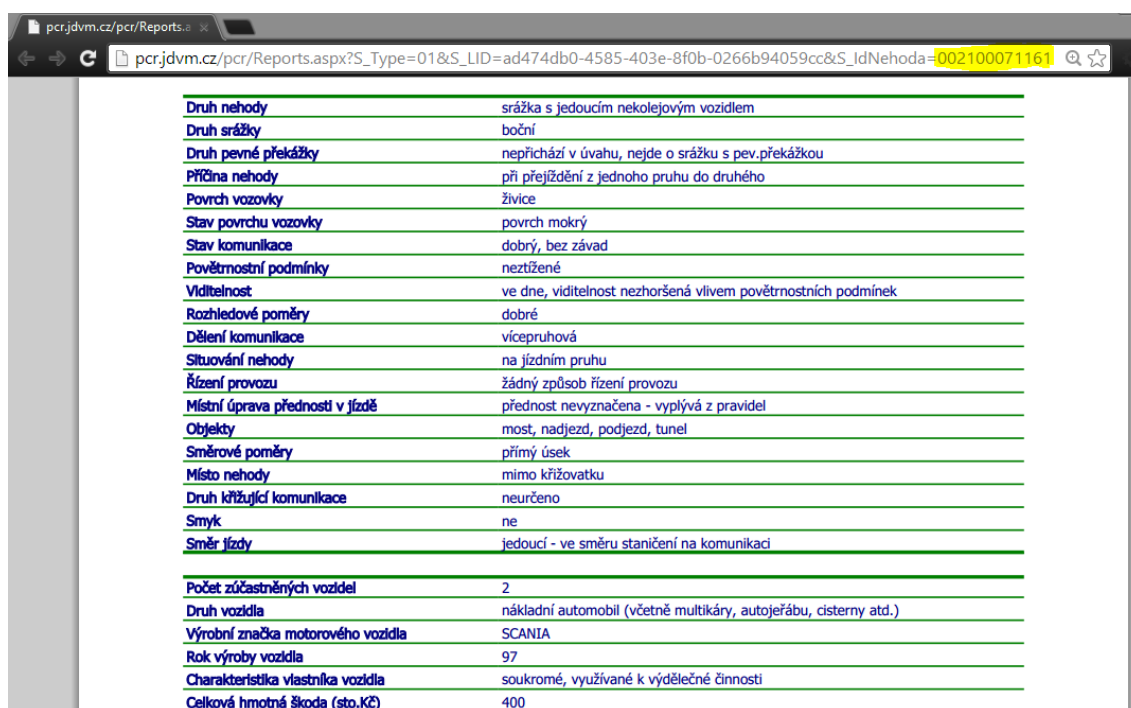
Stahování dat probíhalo právě na tomto serveru, protože disponuje lepší konektivitou a je neustále spuštěný, takže byla data postahována rychleji než při použití lokálního PC. Na serveru jsem využil možnost spouštět skripty psané v jazycích PHP a BASH, databázi MySQL [15] a možnost nastavení plánovaných úloh, tedy démona CRON [16].

### 3 Zdroje dat

Zdrojem informací o dopravních nehodách byla jednotná dopravní vektorová mapa [1]. V tomto systému jsou data dostupná několika způsoby. Na adrese [17] je možné dle zadaných parametrů vyhledat jednotlivé nehody, nicméně vyhledat jich je možné maximálně 100. Pokud je ale nehod nalezeno méně, je k dispozici odkaz na základní informativní výpis o nehodě [18]. Například pro nehodu s ID 002100070013 je výpis dostupný na adrese:

[http://pcr.jdvm.cz/pcr/Reports.aspx?S\\_Type=01&S\\_LID=41aa962a-f5bb-4e2b-953d-c56b6ba94b63&S\\_IdNehoda=002100070013](http://pcr.jdvm.cz/pcr/Reports.aspx?S_Type=01&S_LID=41aa962a-f5bb-4e2b-953d-c56b6ba94b63&S_IdNehoda=002100070013)

V tomto výpisu je ke každé dopravní nehodě uvedeno celkem 44 parametrů a je to nejpodrobnější výpis, který je v této aplikaci k dispozici.



Druh nehody	srážka s jedoucím nekalajícím vozidlem
Druh srážky	boční
Druh pevné překážky	nepřichází v úvahu, nejde o srážku s pev.překážkou
Příčina nehody	při přejíždění z jednoho pruhu do druhého
Povrch vozovky	živice
Stav povrchu vozovky	povrch mokrá
Stav komunikace	dobrý, bez závad
Povětrnostní podmínky	neztížené
Viditelnost	ve dne, viditelnost nezhoršená vlivem povětrnostních podmínek
Rozhledové poměry	dobré
Dělení komunikace	vícepruhová
Situování nehody	na jízdním pruhu
Řízení provozu	žádný způsob řízení provozu
Místní úprava přednosti v jízdě	přednost nevyznačena - vyplývá z pravidel
Objekty	most, nadjezd, podjezd, tunel
Směrové poměry	přímý úsek
Místo nehody	mimo křižovatku
Druh křižující komunikace	neurčeno
Smyk	ne
Směr jízdy	jedoucí - ve směru staničení na komunikaci
Počet zúčastněných vozidel	2
Druh vozidla	nákladní automobil (včetně multikáry, autojeřábu, cisterny atd.)
Výrobní značka motorového vozidla	SCANIA
Rok výroby vozidla	97
Charakteristika vlastního vozidla	soukromé, využívané k výdělečné činnosti
Celková hmotná škoda (sto.Kč)	400

Obrázek 3.1: Základní informativní výpis o nehodě

Dále je k dispozici vyhledávání v mapě [19]. Zde je možné jedním vyhledáním získat celkem 10000 výsledků, výsledky jsou ale stránkovány a aplikace je určena spíše k získání přehledu o lokaci nalezených nehod.

## 4 Získání dat

V této kapitole popisuji celý proces stažení dat. Data budu stahovat z serveru ministerstva dopravy pomocí několika skriptů. Zároveň budu stažená data průběžně konvertovat do jednotné podoby. Výsledkem procesu bude CSV soubor s daty připravenými pro import do databáze. Další operace s daty již budou probíhat právě v databázi.

Uložení dat do souboru CSV umožní snadné zpracování dat v libovolné aplikaci. Takto strukturovaná data nejsou vázána na jeden konkrétní nástroj, ale je možné je využít například v tabulkovém procesoru, importovat do SQL databáze, NoSQL databáze nebo specializovaného dataminigového nástroje. V případě importu do relační databáze je však třeba počítat s tím, že data nejsou v normalizované formě.

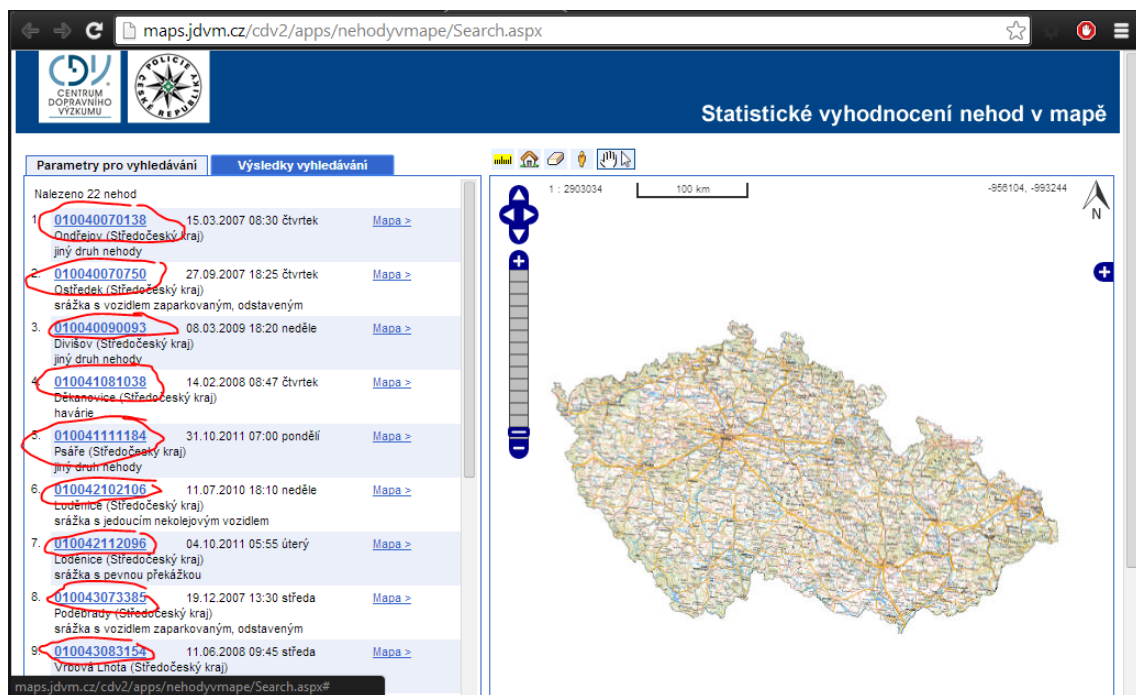
### 4.1 Stažení identifikátorů nehod

Jako hlavní zdroj dat o nehodě jsem použil základní informativní výpis o nehodě [18]. Po zadání parametru `IdNehoda` je vygenerován výpis ve formátu PDF, který je možné uložit a dále zpracovat. Vygenerování výpisu trvá přibližně 11 vteřin a k dispozici jsou nehody od roku 2007, podle oficiálních statistik PČR tedy celkem přibližně 700 tisíc evidovaných nehod [20]. K získání výpisu všech nehod je tedy třeba zjistit více ID nehod.

Identifikátory nehod jsem získal na stránce Statistické vyhodnocení nehod v mapě [19]. Při zadání dotazu bez omezení, tedy na nehody se všemi parametry od 1. 1. 2017, vrátí aplikace jako výsledek informaci, že zadanému dotazu vyhovuje přibližně 700 000 nehod. Z toho jich zobrazí maximálně 10 000 s tím, že stránkování probíhá po 100 záznamech na stránku. Za předpokladu, že dotaz rozdělím na poddotazy tak, aby byl vždy výsledek maximálně 100 záznamů, bude možné stránky s jednotlivými ID nehod stahovat automaticky a následně tyto soubory dále zpracovat.

Dotazy jsem se rozhodl rozdělit po jednotlivých dnech, tedy při stahování bych postupně zvyšoval den a pro ten získal odpovídající nehody. Toto dělení ale nebude dostatečně jemné – za roky 2007 až 2013 by to znamenalo maximálně 255500 stažených nehod, což je přibližně třetina všech dostupných nehod. Dále jsem se rozhodl dotaz pro každý den rozdělit dle jednotlivých obcí ČR. K tomu bylo třeba získat číselník obcí České Republiky. Ten je k dispozici online [21]. Použil jsem sloupec s identifikátory obcí a uložil jej jako prostý textový soubor `cities`.

Dále je třeba odchytnout dotaz, na jehož základě server vygeneruje výsledek. K tomu jsem využil rozšíření Request Maker [22] webového prohlížeče Google Chrome.



Obrázek 4.1: Seznam nehod v mapě

Pro znovuprovedení dotazu jsem použil Chrome DevTools, které jsou součástí webového prohlížeče Google Chrome. Zde na záložce Network je možné prohlížet proběhlé dotazy a zkopírovat do schránky odpovídající příkaz pro znovuprovedení dotazu pomocí nástroje curl [23]. Tento příkaz vypadá následovně:

```
curl -s 'http://maps.jdvm.cz/cdv2/apps/nehodyvmapa/Search.aspx'
-H 'Origin: chrome-extension://kajfghlhfkcoafkcjlajldicbikpgnp'
-H 'Accept-Encoding: gzip,deflate,sdch'
-H 'Host: maps.jdvm.cz'
-H 'Accept-Language: cs-CZ,cs;q=0.8'
-H 'User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/30.0.1599.101 Safari/537.36'
-H 'Content-Type: application/x-www-form-urlencoded'
-H 'Accept: */*'
-H 'Cookie: __utma=201653578.1073276052.1383168085.1383168085.1383168085.1;
__utmc=201653578; __utmz=201653578.1383168085.1.1.utmcsr=policie.cz|
utmccn=(referral)|utmcmd=referral|utmctt=/clanek/dopravni-nehody-
v-mape-cr.aspx;ASP.NET_SessionId=o5pgr2npdxwledvwijnest34'
-H 'Connection: keep-alive'
--data 'TsmMain=UppSearch%7CTcoMain%24TpaObject%24BtnSearch&TcoMain
%24TpaObject%24TxtDatumDo=' $to' &TcoMain%24TpaObject%24MeeDatum
Do_ClientState=&TcoMain%24TpaObject%24DdlDruhNehody=&TcoMain%24
TpaObject%24DdlAlkohol=&TcoMain%24TpaObject%24DdlViditelnost=&Tco
Main%24TpaObject%24DdlDruhVozidla=&TcoMain%24TpaObject%24Ddl
```



```
PocetVozidelSign=1&TcoMain%24TpaObject%24TxtPocetVozidel=&Tco
Main%24TpaObject%24DdlNaslUsmrSign=1&TcoMain%24TpaObject%
24TxtNaslUsmr=&TcoMain%24TpaObject%24DdlNaslTezceSign=1&Tco
Main%24TpaObject%24TxtNaslTezce=&TcoMain%24TpaObject%24Ddl
NaslLehceSign=1&TcoMain%24TpaObject%24TxtNaslLehce=&TcoMain
%24TpaObject%24DdlZavineni=&TcoMain%24TpaObject%24DdlUnik=&
TcoMain%24TpaObject%24DdlTrSilnice=&TcoMain%24TpaObject%24
TxtCisSilnice=&TcoMain%24TpaObject%24TxtOkres=&TcoMain%24T
```

...

```
BtnSearch=Vyhledat' --compressed
```

Zde se předává nástroji `curl` v parametru `--data` serializované pole parametrů, z nichž jsou důležité `TxtDatumOd`, `TxtDatumDo` a `TxtObec`. Právě podle nich bude dělen dotaz na menší části. Jak je to provedeno je vidět v skriptu `get-ids`. Jde o dva vnořené cykly, kdy jeden prochází po řádcích soubor `cities`, ve kterém je na každém řádku uložen kód jedné obce. Druhá smyčka provede průchod po rocích od roku 2007 do roku 2013. Obdobně další dvě vnořené smyčky provedou průchod po měsících a dnech v měsíci. Výsledky vnořené smyčky ukládáme do souborů skládajících se z kódu obce a data, kterému nehody odpovídají. Na konci vnější smyčky prochazející obce ze všech stažených souborů filtruji identifikátory dopravních nehod pomocí standardních unixových nástrojů `cat`, `grep`, `awk` a `cut`. Příkaz `cat` vypisuje zadaný soubor na standardní výstup, příkaz `grep` filtruje řádky odpovídající zadanému řetězci, příkaz `awk` pracuje jako textový filtr a nástroj `cat` filtruje vstup po sloupcích. Tyto nástroje umí číst data ze standardního vstupu a výsledek zapisovat na standardní výstup. Jejich zřetěžením do kolony tak získáme filtr, jehož výstup průběžně přepisují do souboru `ids`.

Propojení standardního vstupu a výstupu je možné pomocí znaku svislá čára (`'|'`, v ASCII tabulce má dekadickou hodnotu 124). Přesměrování standardního výstupu do souboru je možné pomocí znaku `'>'` (v ASCII tabulce má dekadickou hodnotu 62). V konkrétním příkladě získávání identifikátorů nehod předpokládám uloženou webovou stránku v souboru `index.html`. V tomto souboru jsou vždy ID nehod na řádku, na kterém je také `javascript:openDetail(`, za závorkou následuje ID nehody. Výběr těchto řádků zajistí nástroj `grep`, který má jako parametr uvedený textový řetězec. Z každého řádku je užitečný text až za předchozím textovým řetězcem, odfiltrován je tedy nástrojem `awk`. ID nehody je zapsáno v prvních jednoduchých uvozovkách, každý řádek je tedy rozdělen nástrojem `cut` a použito je druhé takto vzniklé pole. Nakonec jsou řádky seřazeny, odstraněny případné duplicity a výsledek připsán do souboru `ids`:

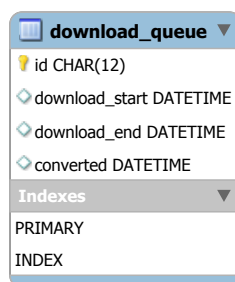
```
cat index.html | grep 'javascript:openDetail(' | \
awk -F'javascript:openDetail' '{print $2}' | cut -d'"' -f2 | \
sort | uniq >> ids
```

Vykonání skriptu `get-ids` tedy provede stažení všech ID dopravních nehod do souboru `ids`. Tento soubor má ale menší počet řádků, než je udáváný celkový počet

nehod serverem. To je způsobeno tím, že se nepodařilo stáhnout všechny ID nehod. Ty se nepodařilo najít, protože informace o nich nebyly zadány dostatečně přesně, například nebyla zadaná poloha, nebylo udáno datum nehody, případně byl udán chybně čas nehody – často byl zadán čas větší než 24 hodin.

## 4.2 Stahování detailů nehod

Samotné stahování detailů nehod bude probíhat na serveru paralelně v několika procesech. Zde je třeba zajistit, aby si procesy vzájemně nepřepisovaly soubory a nestahovaly se duplicitní nehody. To jsem vyřešil vytvořením fronty pomocí databáze MySQL. Databáze obsahuje tabulku `download_queue`.



download_queue	
id	CHAR(12)
download_start	DATETIME
download_end	DATETIME
converted	DATETIME
Indexes	
PRIMARY	
INDEX	

Obrázek 4.2: Schéma tabulky `download_queue`

Tato tabulka má celkem 4 sloupce: ID nehody, datum a časy zahájení stahování, ukončení stahování a proběhlé konverze. Ve výchozím stavu jsou hodnoty nastaveny na NULL [25, strana 103], po provedení příslušné operace se doplní datum a čas jejich vykonání. Tím je zajištěno, že pokud jeden proces aktuálně stahuje určitou nehodu, žádný jiný proces ji již nezačne stahovat. Dále pak získám celkový přehled o provedených staženích a konverzích. Díky tomu je možné například spočítat průměrnou dobu stahování jedné nehody:

```
SELECT AVG(TIMESTAMPDIFF(SECOND, `download_start`, `download_end`))
-> AS `avg_time`
-> FROM `download_queue`;
```

```
+-----+
| avg_time |
+-----+
| 17.2499 |
+-----+
1 row in set (0.94 sec)
```

Stahování nehod probíhá na serveru, konkrétně jej má na starosti skript `download`. Skript je implementován v jazyce PHP a je tvořen jedinou třídou `run`, na konci skriptu je vytvořena instance této třídy. V konstruktoru třídy je provedeno

připojení k MySQL databázi, dále je zavolána metoda `run`. Ta ve smyčce načítá další ID z databáze, u kterého ještě nezačalo stahování. Než začne stahování, nastaví datum a čas začátku stahování pro dané ID. Následně je provedeno uložení staženého souboru do složky pdf. V tuto chvíli je uložen datum a čas dokončení stahování a smyčka se opakuje. Před samotným ukončením skriptu je ještě uzavřeno připojení k databázi.

### 4.3 Konverze stažených dat

Každý PDF soubor s detaily o jedné nehodě je velký přibližně 0,5 MB. Na VPS mám ale k dispozici 20 GB volného místa, stažené soubory tedy průběžně konvertuji do textové podoby. To provádí skript `convert`. Nejprve načte z databáze všechny nehody, které jsou stažené, ale dosud neproběhla konverze. Následně pomocí nástroje `pdftohtml` převede PDF soubor do textové podoby a uloží do proměnné `line`. Tu pomocí kombinace příkazů `grep`, `sed`, `cut`, `tr` a `awk` rozdělím na jednotlivá pole a výsledek průběžně připsuji do výsledného CSV souboru. V tomto souboru je na každém řádku záznam o jedné nehodě a jeho výsledná velikost je asi 700 MB. Se souborem v tomto formátu a v této velikosti je již možné dále pohodlně pracovat.

Jako poslední zbývá zajistit průběžné spouštění skriptu `convert` a v případě, že by byl ukončen skript `download`, tak i jeho spouštění. To je zajištěno skriptem `cron-download-convert`. Ten v případě, že je možné vytvořit další připojení k databázi (limit je nastaven na maximálně 18 souběžných připojení) spustí další proces stahující PDF soubory. Stejně tak, pokud neprobíhá konverze, spustí skript `convert`. Automatického spouštění tohoto skriptu je docíleno následujícím nastavením plánovače `cron`. Plánovač se nastavuje pomocí nástroje `crontab`, konkrétně příkazem `crontab -e`:

```
*/5 * * * * cd /path/to/users/scripts/; ./cron-download-convert
```

Po zadání příkazu se otevře konfigurační soubor plánovače ve výchozím textovém editoru. Po uložení souboru se změny rovnou provedou, není tedy třeba plánovač restartovat. Zda je cron nastaven korektně lze ověřit příkazem `crontab -l`.

Pro kontrolu stavu stažených a zkonvertovaných nehod je k dispozici skript `status`. Ten zobrazí počty stažených a překonvertovaných souborů dle záznamů v databázi, zjistí počet běžících procesů pomocí nástroje `ps` a zobrazí velikost stažených dat. Po spuštění skriptu produkuje následující výstup:

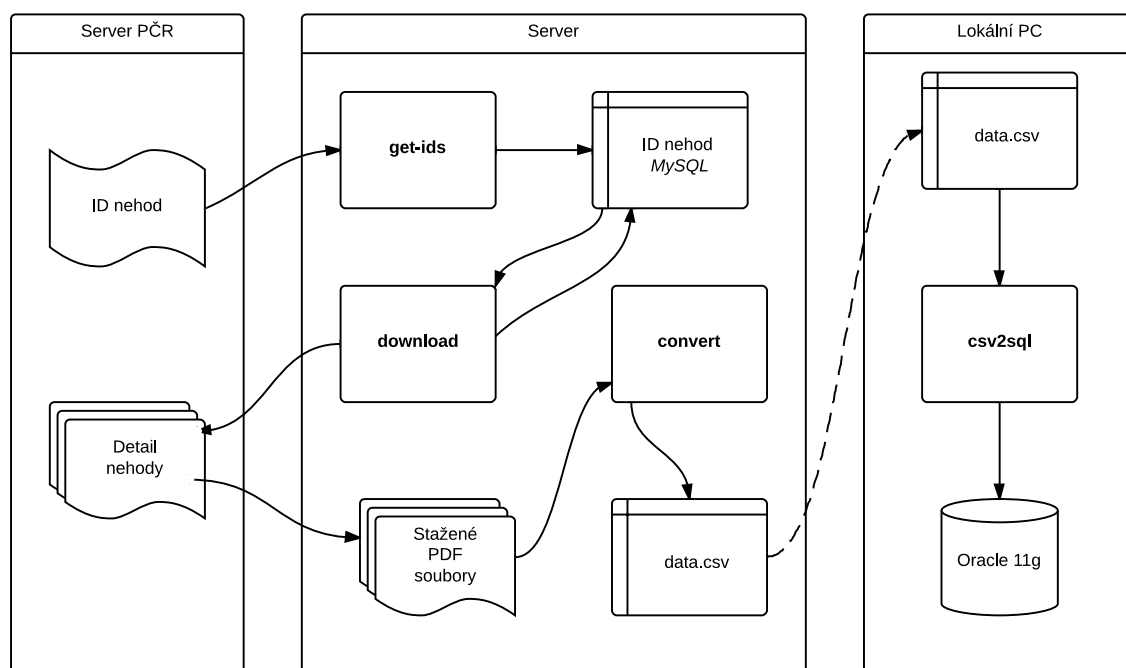
```
user@gentoo ~/nehody $ ./status
--- Mon December 16 09:10:37 CEST 2013 ---
Pocet bezicich stahovacich skriptu: 18
Pocet bezicich konverznich skriptu: 0
Pocet stazenych pdf za posledni minutu: 35
Pocet stazenych pdf za posledni hodinu: 2047
Pocet stazenych pdf za posledni den: 49982
Pocet stazenych pdf celkem: 318524
```

Pocet prevedenych nehod: 318498

Velikost prevedenych dat: 324M

Z výpisu je zřejmé, jakou rychlostí se data stahovala. Při spuštění pouze jednoho procesu, který nehody stahoval sice trvalo stažení jedné nehody přibližně poloviční dobu, nicméně při spuštění více procesů se doba stažení neprodlužovala úměrně vzhledem k počtu paralelních stahování. Jako nejrychlejší se tedy jevilo spustit maximální počet těchto procesů, což vzhledem možnému k počtu připojení na MySQL databázi znamenalo 18 paralelně běžících procesů.

Na následujícím obrázku je schematicky znázorněn celý proces stahování nehod. Je zde vidět, co se odehrává na kterém stroji a kde jsou jednotlivá data uložena.



Obrázek 4.3: Proces stahování nehod

## 5 Návrh databáze

V tuto chvíli již znám strukturu stažených dat a na jejich základě je možné navrhnout podobu databáze. Návrh provedu v nástroji Oracle SQL Developer Data Modeler. Nejprve vytvořím logický model [26, strana 104], ten převedu na relační model (nebo také fyzický datový model [26, strana 110]), ze kterého vygeneruji DDL skript. Po spuštění tohoto skriptu budou vytvořeny prázdné tabulky, do kterých bude možné data uložit. Proběhne také vytvoření definovaných pohledů a indexů.

### 5.1 Struktura zdrojových dat

Data jsou ve zdrojovém CSV souboru uložena tak, že každý řádek reprezentuje jednu nehodu a řádky jsou pomocí středníků děleny do sloupců. Sloupce jsou pak následující:

- ID: 12místný identifikátor dopravní nehody
- Lokalita: Město, kraj
- Datum a čas nehody
- Druh pozemní komunikace: Místní, účelová atd.
- Číslo pozemní komunikace
- Zavinění nehody: Řidičem, chodcem
- Alkohol: Množství alkoholu v promilích
- Usmrceno, lehce zraněno, těžce zraněno osob: Celé číslo
- Druh nehody: Srážka s vozidlem, s pevnou překážkou
- Druh srážky: Boční, zezadu
- Druh pevné překážky: Svodidlo, patník
- Příčina nehody: Nevěnování se řízení vozidla, nedodržení bezpečné vzdálenosti
- Povrch vozovky: Živice, dlažba
- Stav povrchu vozovky: Suchý, mokrý

- Viditelnost: Ve dne, v noci
- Rozhledové poměry: Dobré
- Dělení komunikace: Dvoupruhová, třípruhová, žádné
- Situování nehody: Na jízdním pruhu, na odbočovacím pruhu
- Řízení provozu: Světelná signalizace, místní úprava
- Místní úprava přednosti v jízdě: Dopravními značkami, přednost zprava
- Objekty: Přejezd, parkoviště
- Směrové poměry: Příčný úsek, křižovatka, kruhový objezd
- Místo nehody: Na křižovatce, mimo křižovatku
- Druh křižující komunikace
- Smyk: Ano, ne
- Směr jízdy: Jedoucí ve směru, v protisměru
- Počet zúčastněných vozidel: Celé číslo
- Druh vozidla: Osobní, nákladní
- Výrobní značka vozidla: Škoda, Ford, Opel atd.
- Rok výroby vozidla: Celé číslo
- Charakteristika vlastníka vozidla: Soukromé, podnikatel
- Celková hmotná škoda: Celé číslo
- Škoda na vozidle: Celé číslo
- Vozidlo po nehodě: Došlo k požáru, nedošlo k požáru, žádná z možností
- Únik hmot: Ano, ne
- Způsob vyproštění osob: Nebylo třeba užít násilí, žádné
- Kategorie řidiče: S řidičským oprávněním skupiny B, C
- Stav řidiče: Pod vlivem alkoholu, jiný nepříznivý stav
- Vnější ovlivnění řidiče: Vyhýbání zvěři, žádné ovlivnění

## 5.2 Normalizace databázového systému

Na základě struktury stažených dat jsem vytvořil logický model. Každá nehoda je identifikována jednoznačným 12místným identifikátorem ID. Pro jeho uložení využiji datový typ `char` o přesně dané délce 12 znaků. Bylo by možné použít i celočíselný datový typ `NUMBER`, abych ale zachoval kompatibilitu s dvanáctimístnými identifikátory a nemusel uvozující nuly doplňovat v aplikaci, použil jsem `CHAR`. Z hlediska dopadu na rychlost v Oracle 11g má výběr z těchto datových typů minimální vliv, není tedy důvod `CHAR` nepoužít.

Všechny další položky se vztahují k ID nehody, bylo by tedy možné data uložit do jediné tabulky, kde by každý řádek reprezentoval jednu nehodu. Tento řádek vypadá v CSV souboru následovně:

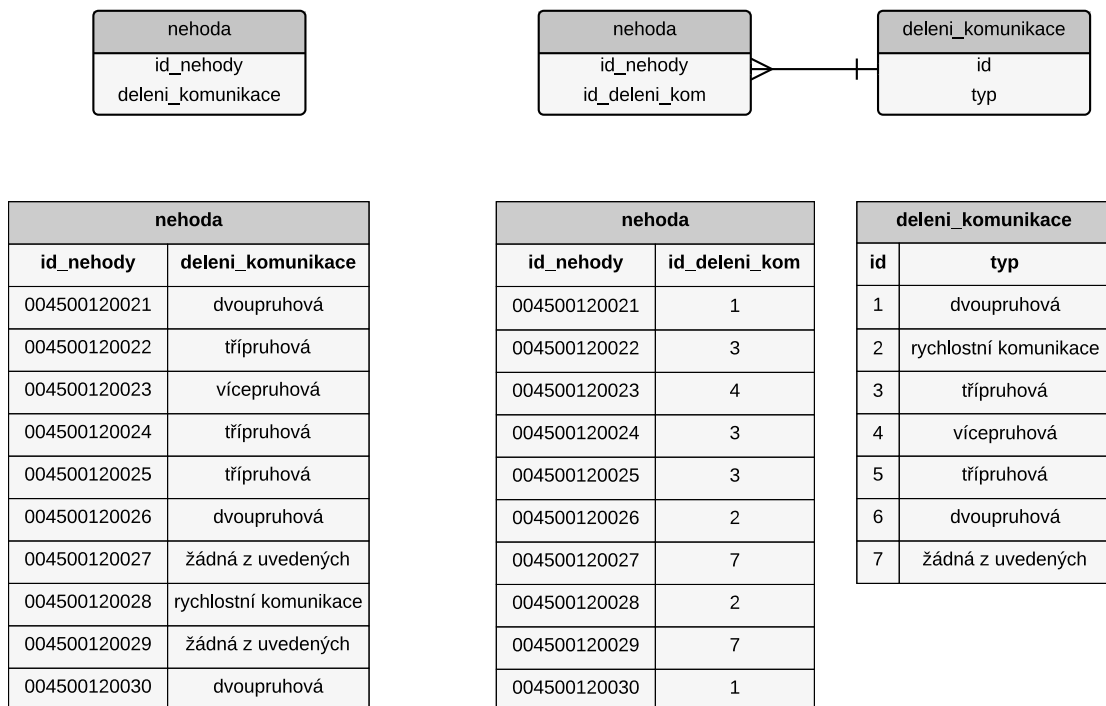
```
002100070013;Praha (Hlavní město Praha);01.01.2007;pondělí;19:00;
komunikace účelová - ostatní (parkoviště apod.);0;řidičem motorového
vozidla;ano, obsah alkoholu v~krvi do 0,99%;0;0;0;srážka s~vozidlem
zaparkovaným, odstaveným;nepřichází v~úvahu, nejde o srážku
jedoucích vozidel;nepřichází v~úvahu, nejde o srážku s~pev.překážkou;
nesprávné otáčení nebo couvání;živice;povrch mokrý;dobrý, bez závad;
neztížené;v noci - s~veřejným osvětlením,viditelnost nezhoršená vlivem
povětrnostních;dobré;žádná z~uvedených;žádné z~uvedených;žádný
způsob řízení provozu;žádná místní úprava;parkoviště přiléhající
ke komunikaci;přímý úsek;mimo křižovatku;neurčeno;ne;vozidlo
jedoucí - na~komunikaci bez staničení;4;osobní automobil bez přívěsu;
ŠKODA;98;soukromé, nevyužívané k~výdělečné činnosti;450;130;
nedošlo k~požáru;žádné z~uvedených;nebylo třeba užít násilí;
s řidičským oprávněním skupiny b;pod vlivem alkoholu, obsah alkoholu
v krvi do 0,99%;řidič nebyl ovlivněn
```

Po prozkoumání dalších řádků je vidět, že některé sloupce nabývají hodnot z přesně definovaného výčtu. Seznam přípustných hodnot pro daný sloupec v CSV souboru jsem získal následujícím příkazem (v příkladu pracuji s desátým sloupcem):

```
cat data.csv | cut -d';' -f10 | sort | uniq
```

Zde se vyskytují celkem tři druhy atributů. Zaprvé to byly atributy, které nabývaly hodnoty `ano` nebo `ne` (případně `nezadáno`). V tomto případě je vhodné použít pro uložení datový typ `Boolean` s možností zadání hodnoty `NULL`. Příkladem takového atributu je `smyk`, ke kterému buď došlo nebo ne. Druhou skupinou jsou sloupce, které nabývají celočíselných hodnot, například celková hmotná škoda, nebo počet zraněných. Poslední skupinou jsou atributy nabývající hodnot z daného výčtu textových položek. Zde jsem využil schéma číselníku [24]. Pro každý takový atribut jsem tedy použil zvláštní tabulku a jednotlivé možnosti výčtu v této tabulce opatřil identifikátory. V hlavní tabulce pak pracuji s těmito identifikátory.

Pro tento účel disponují některé databáze datovým typem `ENUM`. Ten však databáze Oracle nenabízí, bylo by možné namísto něj použít omezení `CHECK`. To by



Obrázek 5.1: Ukázka vytvoření číselníku na ERA diagramu a konkrétních datech

znamenalolo uložení hodnot jako textových řetězců a při uložení kontrolovat, zda je vkládaná hodnota z daného výčtu. Vyčlenění číselníků do zvláštních tabulek je ale vhodnější z několika důvodů. Při použití dat v DM bude možné pracovat s celočíselnými identifikátory, které jsou kratší a zpracování tak bude rychlejší. Dále budou číselníky přehledně k dispozici – pro jejich prohlížení není třeba zobrazovat SQL skript definující tabulku, ale bude stačit zobrazit odpovídající tabulky. Konečně také bude toto řešení výkonnější v případě, kdy pracujeme s denormalizovanými daty, ale zajímají nás jen některé atributy.

### 5.3 Naplnění databáze daty

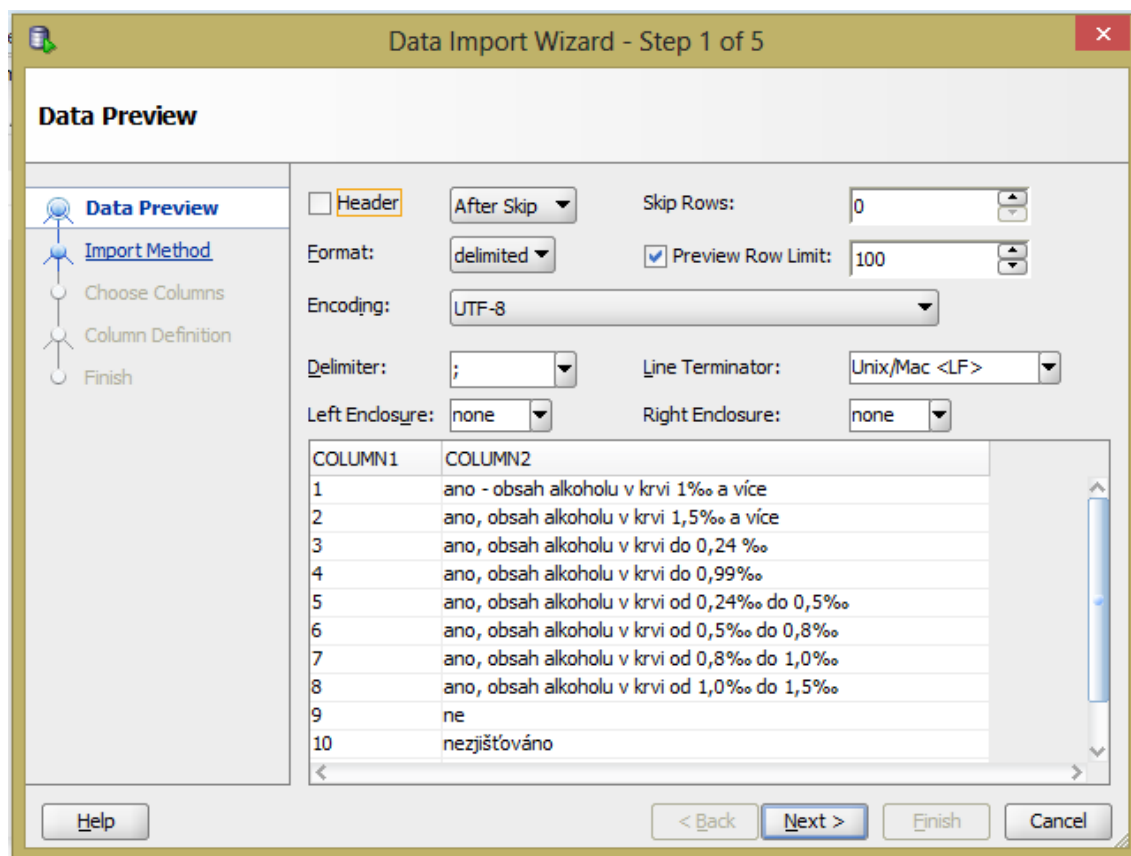
Po spuštění DDL skriptu se vytvoří prázdné tabulky, které zbývá naplnit daty. Nejprve jsem vytvořil číselníky. K jejich vytvoření jsem použil příkaz z předchozí kapitoly, doplněný o přidání čísel řádků, které budou jednoznačnými identifikátory jednotlivých položek.

```
cat data.csv | cut -d';' -f15 | sort | uniq | nl -w1 -s";" \
> Druh_pevne_prekazky.csv
```

Výsledné soubory jsou k dispozici v příloze ve složce enums. Uloženy jsou jako textové soubory s dvěma sloupci oddělenými středníkem.



Import těchto souborů do jednotlivých tabulek jsem provedl v prostředí Oracle SQL Developeru. Ten umožňuje nastavit podrobné parametry importu přesně podle struktury zdrojového souboru. V mém případě šlo o textový soubor s kódováním UTF-8, s dělicím znakem středníkem a unixovými konci řádků (LF). Ostatní nastavení jsem ponechal ve výchozím stavu, tedy bez přeskakování řádků s hlavičkou a jednotlivé buňky bez ohraničení.



Obrázek 5.2: Import dat do databáze z textových souborů

Posledním krokem byl import do hlavní tabulky `dopravni_nehoda`. Zde bylo třeba jednotlivé sloupce zdrojového souboru data.csv přeložit dle číselníků na odpovídající číselné hodnoty. Dále pak bylo třeba u celočíselných hodnot odstranit mezery mezi řády, ošetřit správný formát data a času a vhodně převést booleovská pole na hodnoty `True` a `False`, případně prázdné hodnoty převést na `NULL`.

Toto provádí skript `csv2sql`. Je napsaný v jazyce PHP a obsahuje jednu třídu. Na konci skriptu je vytvořena instance této třídy a následně zavolány metody, do kterých jsou předány parametry příkazové řádky. Skript přejímá celkem dva parametry, prvním je cesta k číselníkům, které jsou umístěny ve složce `enums`. Název této složky je předán metodě `loadTranslations`, která načte všechny soubory v této složce a na jejich základě si vytvoří překladové tabulky. Druhým je název zdrojového CSV souboru s daty, v mém případě `data.csv`, který je následně předán do metody `processFile`, která provede samotné zpracování vstupního souboru. V této metodě

je zdrojový soubor procházen po řádcích, každý řádek je poté zpracován v metodě `processLine`. Každý řádek je rozdělen středníky do pole, jehož prvky jsou buď převedeny na číselné hodnoty metodou `toNumber`, nebo přeloženy dle číselníků metodou `translate`.

Takto zpracované prvky pole jsou poté spojeny zpět do textového řetězce, který je včleněn do výsledného příkazu v jazyce SQL. Výstup skriptu je příkazem `echo` vytištěn na standardní výstup, který je možné následně směřovat do souboru. Výsledný SQL příkaz, který vloží záznam o jedné dopravní nehodě do databáze může vypadat následovně:

```
INSERT INTO DOPRAVNI_NEHODA
  (ID_NEHODY, MESTO_ID, DATUM, POZEMNI_KOMUNIKACE_CISLO,
  ZAVINENI_NEHODY_ID, ALKOHOL_ID, USMRCENO_OSOB, TEZCE_ZRANENO_OSOB,
  LEHCE_ZRANENO_OSOB, DRUH_NEHODY_ID, DRUH_SRAZKY_ID,
  DRUH_PEVNE_PREKAZKY_ID, PRICINA_NEHODY_ID, POVRCH_VOZOVKY_ID,
  STAV_POVRCHU_VOZOVKY_ID, STAV_KOMUNIKACE_ID,
  POVETRNOTSTNI_PODMINKY_ID, VIDITELNOST_ID, ROZHLEDOVE_POMERY_ID,
  DELENI_KOMUNIKACE_ID, SITUOVANI_NEHODY_ID, RIZENI_PROVOZU_ID,
  MISTNI_UPRAVA_PREDNOSTI_ID, OBJEKTY_ID, SMEROVE_POMERY_ID,
  MISTO_NEHODY_ID, DRUH_KRIZUJICI_KOMUNIKACE_ID, SMYK, SMER_JIZDY_ID,
  POCET_ZUCASTNENYCH_VOZIDEL, DRUH_VOZIDLA_ID,
  VYROBNI_ZNACKA_VOZIDLA_ID, ROK_VYROBY_VOZIDLA,
  CHARAKT_VLASTNIKA_VOZIDLA_ID, CELKOVA_HMOTNA_SKODA,
  SKODA_NA_VOZIDLE, VOZIDLO_PO_NEHODE_ID, UNIK_HMOT_ID,
  ZPUSOB_VYPROSTENI_OSOB_ID, KATEGORIE_RIDICE_ID, STAV_RIDICE_ID,
  VNEJSI_OVLIVNENI_RIDICE_ID, DRUH_KOMUNIKACE_ID)
VALUES
  ('002100070013', '2715', TO_DATE('01.01.2007 19:00', 'DD.MM.YYYY
  HH24:MI'), '1', '7', '4', '0', '0', '0', '10', '2', '2', '25',
  '6', '7', '1', '5', '4', '1', '7', '10', '4', '6', '2', '5', '1',
  '2', 'N', '5', '4', '13', '98', '1998', '13', '45000',
  '13000', '2', '5', '1', '7', '6', '6', '4');
```

Pro samotné uložení dat stačí spustit vzniklý SQL skript. To je možné v grafickém prostředí SQL Developer, zde však nejprve dojde k jeho načtení, které je vzhledem k velikosti skriptu 681 MB zdlouhavé. Vhodnější je skript spustit po přihlášení v konzolové aplikaci `sqlplus` příkazem:

```
SQL > @insert-data.sql
```

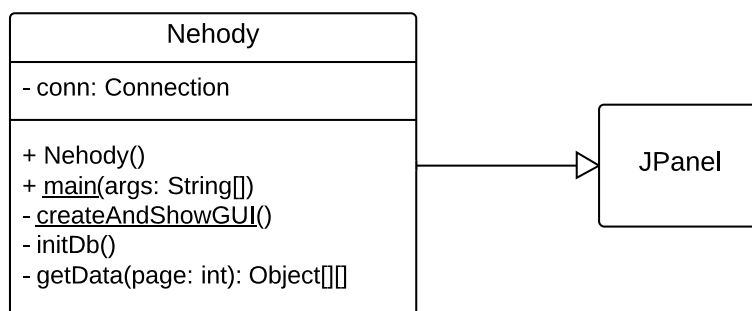
V tuto chvíli je tedy kompletně vytvořena a naplněna daty serverová část DBS. S daty je možné pracovat v libovolné aplikaci, která je schopna se k databázovému serveru připojit, například z konzole pomocí nástroje `sqlplus`, případně nástrojem SQL Developer disponující grafickým uživatelským rozhraním.

## 6 Klientská část databázového systému

Pro možnost prohlížení dat uložených v databázi jsem vytvořil samostatnou aplikaci. Implementována je v jazyce Java, tvořena je jedním oknem a data jsou zobrazena přehledně v tabulce, kterou je možné listovat. Vzhledem k následujícímu zpracování dat není tato aplikace stěžejním bodem práce, disponuje tedy jen základní funkcí.

Aktuálně je aplikace nastavena pro připojení k lokální databázi. URL, na kterou se aplikace připojí je `jdbc:oracle:thin:@localhost:1521:oracle`. Tento řetězec je možné změnit, klientská aplikace se pak může připojovat k databázi běžící na jiném stroji. Pro připojení k databázi se využívá JDBC [25, strana 602] v třídě `DriverManager` dostupné v balíčku `java.sql`, který je součástí instalace JDK.

Struktura aplikace je znázorněna v následujícím diagramu tříd. Ten jsem vytvořil dle notace uvedené v [27, strana 153].



Obrázek 6.1: Diagram tříd klientské aplikace

Aplikace je implementována v jedné třídě `Nehody`, která je potomkem třídy `JPanel`. Vstupním bodem aplikace je metoda `main`, ze které je tvořeno hlavní okno aplikace. To je vytvořeno v metodě `createAndShowGUI`. Toto okno je odvozeno od třídy `JFrame` a obsahuje toolbar s tlačítky pro listování nalezenými dopravními nehodami. Dále obsahuje tabulku, která je naplněna daty z databáze. Připojení k databázi je vytvořeno v metodě `initDb` a uloženo v proměnné `conn`.

Tabulka, která zobrazuje nalezené záznamy je instancí třídy `JTable`. Do záhlaví tabulky je předáno pole atributů jednotlivých nehod a obsah tabulky je pro každou stránku získán z databáze s odpovídajícím offsetem. Databáze Oracle 11g nepodporuje obvyklé příkazy `OFFSET` a `LIMIT`, stránkování je tedy řešeno řádkovým

poddotazem [25, strana 224] s omezením příkazem ROWNUM. Například pro druhou stránku, na které jsou zobrazeny záznamy od padesátého do stého vypadá výsledný SQL dotaz následovně:

```
SELECT * FROM
  (SELECT neh.*, ROWNUM rnum FROM
    (SELECT * FROM nehody ORDER BY id) neh
   WHERE ROWNUM <= 50)
 WHERE rnum >= 50;
```

Pro vývoj aplikace jsem použil vývojové prostředí NetBeans, které automaticky vygenerovalo soubory manifest.mf a build.xml. Dle nich je možné pomocí překladače ant příkazem `ant build` vytvořit spustitelnou aplikaci. Překlad je také možné provést v prostředí NetBeans příkazem `Run Project`, který je dostupný v menu `Run`. Po přeložení projektu se také vytvoří soubor `nehody.jar` ve složce `dist`. Tento soubor je možné přenést na libovolný stroj, který se může k databázi připojit, ke svému běhu však vyžaduje běhové prostředí Java.

Číslo nehody	Město	Datum	Den v týdnu	Čas	Číslo	Způsob zavinění nehody	Alkohol	Usmrceno	Těžce zraněno	Lehce zraněno	Druh nehody	Druh srážky
002100070001		01.01.2007	PONDĚLÍ	02:45	0	řidičem motorového vozidla	ne	0	0	0	srážka s jedoucím nekolejov...	zezadu
002100070002		01.01.2007	PONDĚLÍ	01:25	0	řidičem motorového vozidla	ano, obsah a...	0	1	1	srážka s vozidlem zaparkova...	nepřichází v úvahu
002100070003		01.01.2007	PONDĚLÍ	03:30	0	řidičem motorového vozidla	ne	0	0	1	srážka s pevnou překážkou	nepřichází v úvahu
002100070004		01.01.2007	PONDĚLÍ	04:15	0	řidičem motorového vozidla	ne	0	0	0	srážka s pevnou překážkou	nepřichází v úvahu
002100070005		01.01.2007	PONDĚLÍ	00:00	0	řidičem motorového vozidla	nezjišťováno	0	0	0	srážka s pevnou překážkou	nepřichází v úvahu
002100070006		01.01.2007	PONDĚLÍ	10:30	0	řidičem motorového vozidla	ne	0	0	0	srážka s jedoucím nekolejov...	z boku
002100070007		01.01.2007	PONDĚLÍ	00:00	0	řidičem motorového vozidla	nezjišťováno	0	0	0	srážka s vozidlem zaparkova...	nepřichází v úvahu
002100070008		01.01.2007	PONDĚLÍ	17:15	0	řidičem motorového vozidla	ne	0	0	0	srážka s jedoucím nekolejov...	z boku
002100070009		01.01.2007	PONDĚLÍ	17:35	0	chodcem	nezjišťováno	0	0	0	srážka s chodcem	nepřichází v úvahu
002100070010		01.01.2007	PONDĚLÍ	17:35	0	řidičem motorového vozidla	ne	0	0	0	srážka s vozidlem zaparkova...	nepřichází v úvahu
002100070011		01.01.2007	PONDĚLÍ	00:00	0	řidičem motorového vozidla	nezjišťováno	0	0	0	srážka s vozidlem zaparkova...	nepřichází v úvahu
002100070012		01.01.2007	PONDĚLÍ	00:00	0	řidičem motorového vozidla	nezjišťováno	0	0	0	srážka s vozidlem zaparkova...	nepřichází v úvahu
002100070013	Praha	01.01.2007	PONDĚLÍ	19:00	0	řidičem motorového vozidla	ano, obsah a...	0	0	0	srážka s vozidlem zaparkova...	nepřichází v úvahu
002100070014		01.01.2007	PONDĚLÍ	20:05	0	řidičem motorového vozidla	ne	0	0	0	srážka s jedoucím nekolejov...	z boku
002100070015		01.01.2007	PONDĚLÍ	20:10	0	řidičem motorového vozidla	ne	0	0	0	srážka s pevnou překážkou	nepřichází v úvahu
002100070016		02.01.2007	ÚTERÝ	06:30	0	řidičem motorového vozidla	ne	0	0	0	srážka s jedoucím nekolejov...	zezadu
002100070017		02.01.2007	ÚTERÝ	07:00	0	řidičem motorového vozidla	ne	0	0	0	srážka s jedoucím nekolejov...	z boku
002100070018		02.01.2007	ÚTERÝ	07:05	0	řidičem motorového vozidla	ne	0	0	0	srážka s pevnou překážkou	nepřichází v úvahu
002100070019	Praha	02.01.2007	ÚTERÝ	07:30	0	řidičem motorového vozidla	ne	0	0	0	srážka s pevnou překážkou	nepřichází v úvahu
002100070020		02.01.2007	ÚTERÝ	00:00	0	řidičem motorového vozidla	nezjišťováno	0	0	0	srážka s vozidlem zaparkova...	nepřichází v úvahu
002100070021		01.01.2007	PONDĚLÍ	00:00	0	řidičem motorového vozidla	nezjišťováno	0	0	0	srážka s vozidlem zaparkova...	nepřichází v úvahu
002100070022	Praha	02.01.2007	ÚTERÝ	08:30	0	řidičem motorového vozidla	ne	0	0	0	srážka s jedoucím nekolejov...	boční
002100070023	Praha	02.01.2007	ÚTERÝ	09:00	0	řidičem motorového vozidla	ne	0	0	0	srážka s jedoucím nekolejov...	z boku
002100070024		02.01.2007	ÚTERÝ	08:30	0	řidičem motorového vozidla	ne	0	0	0	srážka s jedoucím nekolejov...	z boku
002100070025		02.01.2007	ÚTERÝ	08:55	0	řidičem motorového vozidla	ne	0	0	0	srážka s vozidlem zaparkova...	nepřichází v úvahu
002100070026	Praha	02.01.2007	ÚTERÝ	08:40	0	řidičem motorového vozidla	ne	0	0	0	srážka s jedoucím nekolejov...	zezadu
002100070027	Praha	02.01.2007	ÚTERÝ	09:25	0	řidičem motorového vozidla	ne	0	0	0	srážka s jedoucím nekolejov...	zezadu
002100070028	Praha	02.01.2007	ÚTERÝ	10:05	0	řidičem motorového vozidla	ne	0	0	0	srážka s jedoucím nekolejov...	boční
002100070029	Praha	02.01.2007	ÚTERÝ	10:30	0	řidičem motorového vozidla	nezjišťováno	0	0	0	srážka s pevnou překážkou	nepřichází v úvahu
002100070030		02.01.2007	ÚTERÝ	00:00	0	řidičem motorového vozidla	nezjišťováno	0	0	0	srážka s vozidlem zaparkova...	nepřichází v úvahu
002100070031	Praha	02.01.2007	ÚTERÝ	10:50	0	řidičem motorového vozidla	ne	0	0	0	srážka s pevnou překážkou	nepřichází v úvahu
002100070032		02.01.2007	ÚTERÝ	10:55	0	řidičem motorového vozidla	ne	0	0	0	srážka s jedoucím nekolejov...	zezadu
002100070033	Praha	02.01.2007	ÚTERÝ	11:45	0	řidičem motorového vozidla	ne	0	0	0	srážka s vozidlem zaparkova...	nepřichází v úvahu
002100070034		02.01.2007	ÚTERÝ	11:00	0	řidičem motorového vozidla	ne	0	0	0	srážka s jedoucím nekolejov...	zezadu
002100070035	Praha	02.01.2007	ÚTERÝ	11:40	0	řidičem motorového vozidla	ne	0	0	0	srážka s jedoucím nekolejov...	zezadu
002100070036	Praha	02.01.2007	ÚTERÝ	11:08	0	řidičem motorového vozidla	ne	0	0	0	srážka s pevnou překážkou	nepřichází v úvahu
002100070037	Praha	02.01.2007	ÚTERÝ	12:15	0	řidičem motorového vozidla	ne	0	0	0	srážka s vozidlem zaparkova...	nepřichází v úvahu
002100070038	Praha	02.01.2007	ÚTERÝ	12:40	0	řidičem motorového vozidla	ne	0	0	0	srážka s jedoucím nekolejov...	boční

Obrázek 6.2: Klientská aplikace zobrazující detaily dopravních nehod

Pro zobrazení dat v aplikaci jsem v databázi vytvořil pohled [25, strana 379], `NEHODY`. Tento pohled vychází z tabulky `DOPRAVNI_NEHODA`. Na ni jsou napojeny všechny číselníky a datum je rozděleno na dny v týdnu, měsíce a roky, aby podle nich bylo možné nehody klasifikovat. Dále jsou binární hodnoty převedeny do textové podoby. Aby byly zobrazeny i řádky, které nemají zadanou některou hodnotu z číselníků, napojuji tabulky příkazem `LEFT JOIN`. Tento pohled je nejvíce denormalizovanou podobou uložených dat.

## 7 Využití dat v dataminingové studii

Hlavním důvodem získání dat o dopravních nehodách bylo jejich využití v dataminingové studii. Bude tedy proveden proces vyhledávání znalostí z dat (KDD) [11, strana 32], který povede k nalezení netriviálních souvislostí v datech, vytvoření modelů, reprezentujících nalezené vzory, a následnou interpretaci těchto modelů.

V této kapitole uvedu příklady konkrétních úloh, které by bylo možné řešit. Primárně předpokládám řešení přímo s daty uloženými v databázi, je však možné použít libovolný jiný nástroj. Pro tento účel jsou k dispozici data v univerzálním formátu CSV dostupná v příloze ve složce data.

### 7.1 Nejčastější příčiny tragických nehod

Cílem této úlohy je prozkoumání možných příčin nehod uložených v databázi, nalezení skupin typických nehod a následná lokalizace výskytu těchto nehod dle okresů a denní doby pro navržení vhodného rozmístění preventivních policejních hlídek. Nejprve tedy bude třeba vybrat atributy, s nimiž se bude v úloze pracovat. v úvahu přichází:

- Alkohol
- Zavinění nehody
- Vnější ovlivnění řidiče
- Viditelnost
- Stav povrchu vozovky
- Stav komunikace
- Směrové poměry
- Rozhledové poměry
- Příčina nehody
- Povrch vozovky
- Povětrnostní podmínky

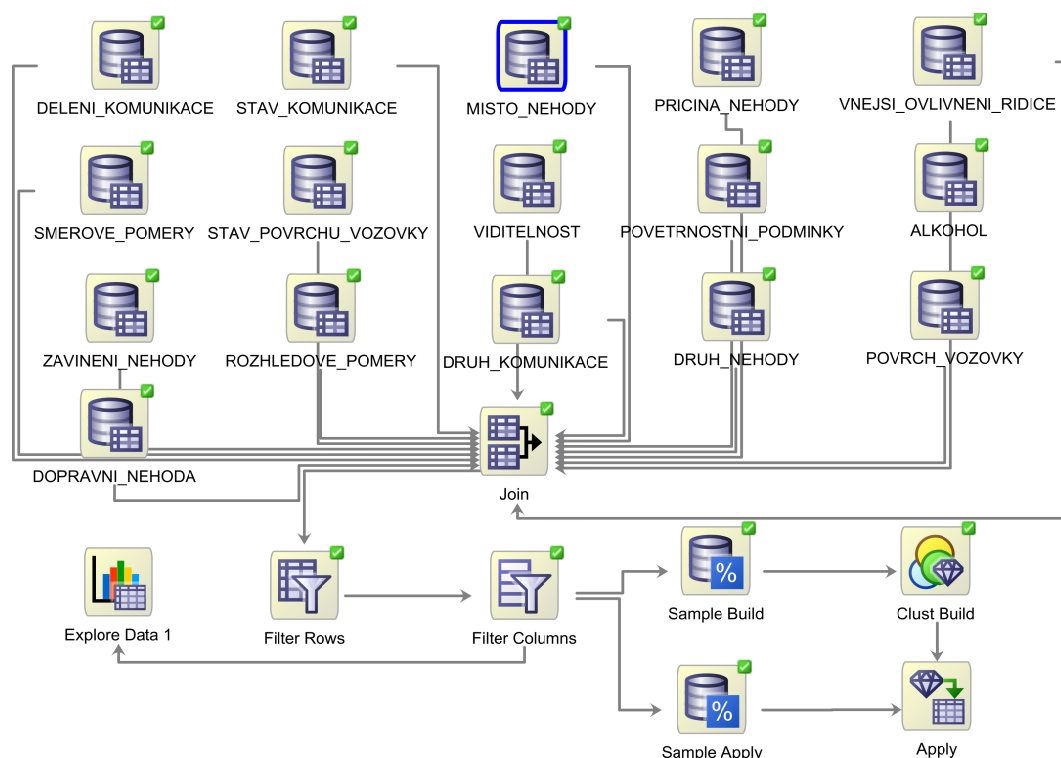
- Místo nehody
- Druh nehody
- Druh komunikace
- Dělení komunikace

Dále bude třeba ze zkoumaného vzorku dat odebrat ty, které nemají zadány požadované vlastnosti, abychom pracovali jen s dostatečně kvalitními daty. Odfiltrovat lze také nehody, u nichž nedošlo k žádným zraněním a nejsou tedy vedeny jako tragické. U připravených dat je možné provést klasifikaci [11, strana 43]. Předpokládá se výběr vhodného algoritmu, výsledkem je nalezení nejčastějších kombinací podmínek, za nichž nehody vznikají.

Posledním krokem bude aplikace nalezených modelů, tedy pro každý z nich nalézt v mapě nejčastější výskyty v jednotlivých okresech. Dále je vhodné hledat výskyty nejen podle okresů, ale také podle časových intervalů, aby byly policejní hlídky na inkriminovaných místech ve správnou dobu.

## 7.2 Řešení úlohy pomocí nástroje Oracle Dataminer

Úlohu z předchozí kapitoly jsem řešil pomocí nástroje Oracle Dataminer. Jde o rozšíření nástroje Oracle SQL Developer a pracuje s daty uloženými v databázi Oracle.



Obrázek 7.1: Vytvořené uzly v Oracle Dataminer

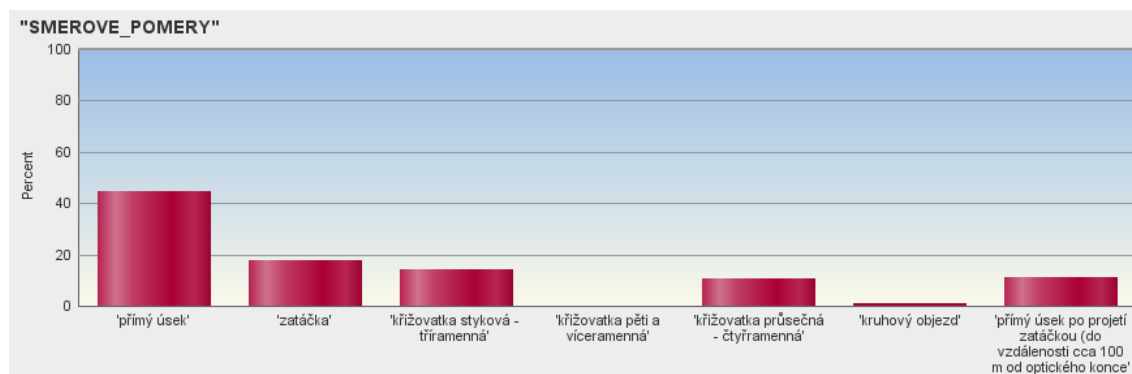
Kromě tohoto nástroje s grafickým uživatelským rozhraním je možné provádět analýzu přímo v jazyce PL/SQL, pro názornost jsem se však rozhodl využít Dataminer. Nejprve jsem transformoval data do podoby, kterou jsem podrobil analýze. Pro tento účel je možné buď vytvořit v databázi pohled, nebo totéž provést přímo v Datamineru.

Rozhodl jsem se pro druhou možnost, následovalo vytvoření uzlů a jejich následné pospojování. V Datamineru jsou pro tento účel uzly **Data Source**. Každý uzel reprezentuje jednu databázovou tabulku, kde je možné vybrat sloupce, které budou použity. Vytvořil jsem tedy uzel **DOPRAVNI\_NEHODA** a dále uzly pro všechny použité číselníky. Ty jsou vedeny do uzlu **Join**, kde je provedeno jejich spojení. Použito je přirozené vnitřní spojení, čímž jsou eliminovány záznamy s hodnotami **NULL**.

Následně jsou v uzlu **Filter Rows** vyfiltrovány záznamy odpovídající těžkým dopravním nehodám, tedy nehodám, ve kterých došlo k těžkému zranění nebo úmrtí. To je zajištěno podmínkou:

`"TEZCE_ZRANENO_OSOB" > 0 OR "USMRCENO_OSOB" > 0`

Vyfiltrovaná data lze prozkoumat po kliknutí na uzel **Explore Data** a následném výběru položky **View Data** z kontextové nabídky. Data je možné zobrazit buď v tabulce nebo v grafu.



Obrázek 7.2: Vizualizace dat v uzlu Explore Data

Nyní je možné data podrobit shlukové analýze. Pro clusterování nabízí dataminer dva algoritmy – K-Means a O-Cluster, v tomto příkladě použiji prvně jmenovanou metodu K-Means.

K-Means [28] je nehierarchický algoritmus třídící data do shluků na základě jejich podobnosti. Při jeho spuštění definujeme počet shluků, který algoritmem chceme získat. Pro výpočet podobnosti prvků je možné použít různých metrik, Dataminer nabízí eukleidovskou, kosinovou a rychlou kosinovou, já použil výchozí eukleidovskou. Vzdálenost dvou bodů v  $n$ -rozměrném prostoru se spočítá pomocí vzorce:

$$d(p, q) = \sqrt{[(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2]} = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$



Dále algoritmus pracuje s centroidy, což jsou prvky reprezentující jednotlivé shluky. Samotný algoritmus probíhá následovně:

1. Pro každý shluk jsou náhodně zvoleny centroidy
2. Každý prvek je přiřazen do shluku náležící nejbližšímu centroidu
3. Přepočtení centroidu
4. Kroky 2. a 3. opakujeme, dokud se nově vypočtený centroid liší od předcházejícího, jinak je algoritmus ukončen

Výsledné shluky jsou v příloze, např. centroid největšího výsledného shluku vypadá následovně:

Centroid	Cluster Rule	
Name	▼ Importance	Mode
SMEROVE_POMERY	0,8000	přímý úsek
MISTO_NEHODY	0,6667	mimo křižovatku
PRICINA	0,5833	řidič se plně nevěnoval řízení vozidla
POVETRNOTSNI_PODMINKY	0,5000	neztížené
DELENI_KOMUNIKACE	0,5000	dvoupruhová
ZPUSOB_ZAVINENI	0,3333	řidičem motorového vozidla
POVRCH_VOZOVKY	0,0000	živice
DRUH_NEHODY	0,0000	srážka s chodcem
DRUH_KOMUNIKACE	0,0000	silnice 1. třídy
ALKOHOL	0,0000	ne

Obrázek 7.3: Centroid největšího shluku

Analýzovaná data jsem rozdělil na dvě poloviny, první jsem použil jako množinu trénovacích dat, na jejichž základě vznikl model. Tento model jsem poté aplikoval na druhou polovinu dat, což proběhlo v uzlu `Apply`. Výsledkem je přiřazení nehod do jednotlivých shluků, což je možné zobrazit, uložit do nové tabulky nebo uložit jako nový sloupec u stávající tabulky dopravních nehod.

## 7.3 Nehody způsobené vysokou rychlostí

Cílem této úlohy je zaměřit se na nehody způsobené nedodržením povolené rychlosti nebo agresivní jízdou. Na základě výsledků by bylo možné rozmístit radary, měřící rychlost vozidel.

V první fázi by bylo vhodné získat vzorek dat odpovídající zadanému problému. Konkrétně by se jednalo o nehody těchto vlastností:

- Zavínění nehody: Řidičem motorového vozidla
- Směrové poměry: Přímý úsek, zatáčka



- Příčina nehody: Bezohledná, agresivní, neohleduplná jízda, jiný druh nepřiměřené rychlosti, nepř. rychlosti bočnímu, nárazovému větru (i při míjení, předjíždění vozidel), nepř. rychlosti dopravně technickému stavu vozovky, nepř. rychlosti stavu vozovky (náledí, výtluky, bláto, mokrá povrch apod.), nepř. rychlosti vlastnostem vozidla a nákladu, nepřizpůsobení rychlosti hustotě provozu, nepřizpůsobení rychlosti viditelnosti (mlha, soumrak, jízda na tlumená světla)

U těchto nehod by bylo vhodné analyzovat veškeré atributy související s místem dopravní nehody. Výsledkem by měl být seznam konkrétních míst, kam umístit radary, případně kam vyslat policejní hlídky měřit rychlost. Znázornit je by bylo možné v tabulkách kde by každý okres obdržel seznam míst na svém území, nebo v mapě.

## 7.4 Nejrizikovější křižovatky

V této úloze by šlo o analýzu křižovatek, kruhových objezdů a železničních přejezdů. Pro každý druh křížení by bylo třeba podobně jako v předchozích úlohách analyzovat příčiny nehod. Cílem by bylo nalezení nejčastějších skupin příčin nehod a na jejich základě navrhnout úpravy konkrétních křížení, například nahrazení některých železničních přejezdů dle skupiny příčin nadjezdy, případně zvýšení zabezpečení instalací závor.

## 7.5 Rizikovost řidičů pro pojišťovny

Cílem úlohy je rozdělit potenciální zájemce o pojištění do skupin dle jejich rizikovosti. Na základě výsledků této úlohy by řidiči s menší pravděpodobností pojistné události (nehody), mohli být zařazeni do skupiny s nižší sazbou pojistného a stejně tak riziková řidiči do opačných skupin. Pro tento účel by mohla být k dispozici podrobnější data, nicméně z aktuálně dostupných atributů by bylo možné analyzovat značku motorového vozidla, stav řidiče (invalida, nemoc, atd.), kategorii řidiče, druh vozidla a charakteristiku vlastníka vozidla.

V první fázi podrobíme data shlukové analýze. Nalezenými shluky mohou být například:

- 1. shluk
  - Značka vozidla: Škoda, Ford
  - Charakteristika vlastníka vozidla: státní podnik, soukromá organizace
  - Kategorie řidiče: s řidičským oprávněním skupiny B
- 2. shluk
  - Charakteristika vlastníka vozidla: Mezinárodní kamionová doprava

- Značka vozidla: Mercedes, Volvo
- Druh vozidla: autobus, nákladní automobil

Nalezené shluky by poté byly rozděleny do skupin dle následků odpovídajících nehod, tedy dle množství usmrcených osob, lehce zraněných osob, těžce zraněných osob, škody na vozidle, celkové hmotné škody, případně úniku hmot a způsobu vyproštění osob. Pravděpodobně bychom tak objevili kombinace charakteristik řidičů, které jsou předpokladem žádných nebo maximálně lehkých dopravních nehod. Stejně tak by mohly být objeveny skupiny atributů, které jsou předpokladem pro pojišťovnu velmi nákladných nehod.

## 8 Závěr

V této bakalářské práci jsem nejprve našel zdroje dat o dopravních nehodách. Dále jsem navrhl způsob, jak data z těchto zdrojů získat a navrženým způsobem data následně postahoval. Při stahování dat jsem zjistil, že u některých nehod nejsou zadány všechny atributy, případně jsou některé z nich zadány chybně. Nekompletně zadané nehody jsem přesto postahoval, protože pro určitý typ analýzy může část zadaných atributů postačovat.

Vzhledem k tomu, že jsem našel dostatečně kvalitní zdroj dat jsem návrh DBS pro uložení záznamů o dopravních nehodách provedl na základě těchto dat. Dle návrhu jsem databázi implementoval v prostředí Oracle 11g a naplnil ji daty. Postahovaná data jsou k dispozici v univerzálním formátu CSV a v normalizované podobě uložené v relační databázi. Pro možnost prohlížení dat uložených v databázi jsem vytvořil klientskou aplikaci v jazyce Java, která zobrazuje data uložená na serveru v databázovém systému.

Na závěr jsem navrhl možné úlohy pro využití dat dataminingovými postupy. Navrhl jsem celkem tři úlohy, jednu z nich jsem vyřešil shlukovou analýzou pomocí nástroje Oracle Dataminer.

## Literatura

- [1] Ministerstvo dopravy. *Jednotná dopravní vektorová mapa* [online]. 2014 [cit. 2014-05-12]. Dostupné z: <http://www.jdvm.cz/>
- [2] Policie České republiky. *Statistika nehodovosti* [online]. 2014 [cit. 2014-05-12]. Dostupné z: <http://www.policie.cz/clanek/statistika-nehodovosti-900835.aspx>
- [3] The National Archives. *Open Government Licence* [online]. 2014 [cit. 2014-05-12]. Dostupné z: [http://en.wikipedia.org/wiki/Open\\_Government\\_Licence](http://en.wikipedia.org/wiki/Open_Government_Licence)
- [4] DATA.GOV.UK. *Road Safety Data* [online]. 2014 [cit. 2014-05-12]. Dostupné z: <http://data.gov.uk/dataset/road-accidents-safety-data>
- [5] NEČASKÝ, Martin. *opendata.cz* [online]. 2014 [cit. 2014-05-12]. Dostupné z: <http://opendata.cz/>
- [6] *Oracle Database 11g* [online]. 2014 [cit. 2014-05-12]. Dostupné z: <http://www.oracle.com/cz/solutions/midsize/oracle-products/database/index.html>
- [7] IBM Corporation. *IBM SPSS Modeler* [online]. 2014 [cit. 2014-05-12]. Dostupné z: <http://www-01.ibm.com/software/analytics/spss/products/modeler/>
- [8] SAS Institute Inc. *SAS Enterprise Miner* [online]. 2014 [cit. 2014-05-12]. Dostupné z: [http://www.sas.com/en\\_us/software/analytics/enterprise-miner.html](http://www.sas.com/en_us/software/analytics/enterprise-miner.html)
- [9] *SAP Business Intelligence* [online]. 2014 [cit. 2014-05-12]. Dostupné z: <http://www.sap.com/pc/analytics/business-intelligence.html>

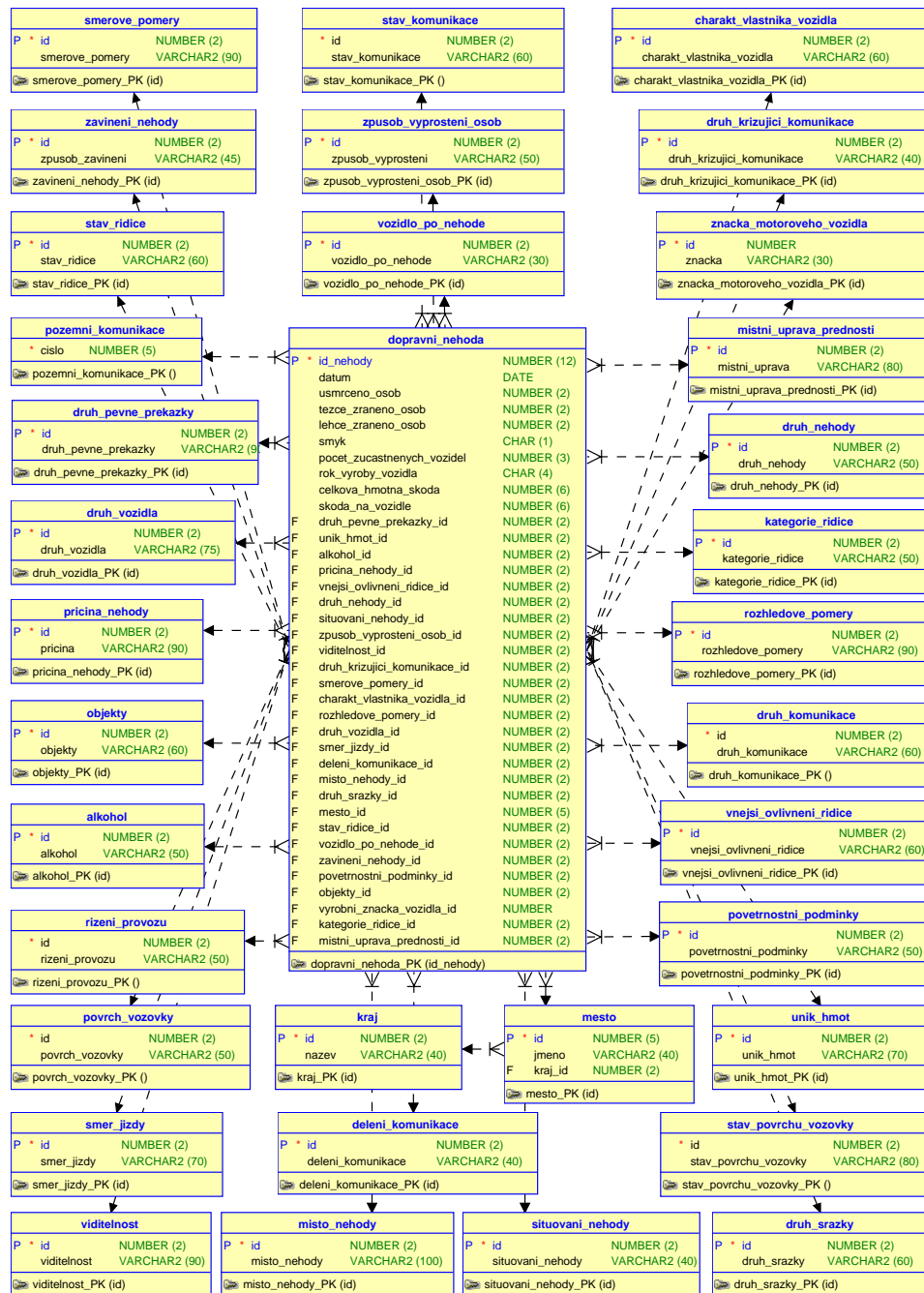
- [10] The R Foundation. *The R Project for Statistical Computing* [online].  
2014 [cit. 2014-05-12]. Dostupné z:  
<http://www.r-project.org/>
- [11] SKALSKÁ, Hana. *Data mining a klasifikační modely*.  
Vyd. 1. Hradec Králové: Gaudeamus, 2010, 153 s. ISBN 978-807-4350-887.
- [12] *Oracle SQL Developer Data Modeler* [online].  
2014 [cit. 2014-05-12]. Dostupné z:  
[http://www.oracle.com/technetwork/developer-tools/  
datamodeler/overview/index.html?ssSourceSiteId=otnru](http://www.oracle.com/technetwork/developer-tools/datamodeler/overview/index.html?ssSourceSiteId=otnru)
- [13] *Oracle SQL Developer* [online].  
2014 [cit. 2014-05-12]. Dostupné z:  
[http://www.oracle.com/technetwork/developer-tools/  
sql-developer/overview/index.html](http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html)
- [14] BRYLA, Bob a LONEY Kevin. *Mistrouství v Oracle Database 11g*  
Vyd. 1. Brno: Computer Press, 2009, 700 s. ISBN 978-80-251-2189-4.
- [15] Oracle. *MySQL* [online].  
2014 [cit. 2014-05-12]. Dostupné z: <http://www.mysql.com/>
- [16] VIXIE, Paul. *cron(8) - Linux man page* [online].  
2014 [cit. 2014-05-12]. Dostupné z: <http://linux.die.net/man/8/cron>
- [17] Policie České republiky. *Dopravní nehody* [online].  
2014 [cit. 2014-05-12]. Dostupné z: <http://pcr.jdvm.cz/pcr/>
- [18] Ministerstvo dopravy České republiky. *Základní informativní výpis o nehodě* [online].  
2014 [cit. 2014-05-12]. Dostupné z:  
[http://pcr.jdvm.cz/pcr/Reports.aspx?S\\_Type=01&S\\_LID=  
41aa962a-f5bb-4e2b-953d-c56b6ba94b63&S\\_IdNehoda=002100070013](http://pcr.jdvm.cz/pcr/Reports.aspx?S_Type=01&S_LID=41aa962a-f5bb-4e2b-953d-c56b6ba94b63&S_IdNehoda=002100070013)
- [19] Policie České republiky. *Statistické vyhodnocení nehod v mapě* [online].  
2014 [cit. 2014-05-12]. Dostupné z:  
<http://maps.jdvm.cz/cdv2/apps/nehodyvmapě/Search.aspx>
- [20] Český statistický úřad. *Kriminalita, nehody* [online].  
2014 [cit. 2014-05-12]. Dostupné z:  
<http://www.czso.cz/x/krajedata.nsf/oblast2/kriminalita-xa>
- [21] Ministerstvo vnitra České republiky. *Databáze adres v ČR a číselníky územních celků* [online].  
2014 [cit. 2014-05-12]. Dostupné z:  
[http://www.mvcr.cz/clanek/database-adres-v-cr-a-ciselniky\  
-uzemnich-celku.aspx](http://www.mvcr.cz/clanek/database-adres-v-cr-a-ciselniky\u-uzemnich-celku.aspx)

- [22] NURMINEN, Juho. *Request Maker* [online].  
2014 [cit. 2014-05-12]. Dostupné z:  
[https://chrome.google.com/webstore/detail/request-maker/  
kajfghlhfkcoafkcjlajldicbikpgnp](https://chrome.google.com/webstore/detail/request-maker/kajfghlhfkcoafkcjlajldicbikpgnp)
- [23] STENBERG, Daniel. *curl and libcurl* [online].  
2014 [cit. 2014-05-12]. Dostupné z:  
<http://curl.haxx.se/>
- [24] Český statistický úřad. *Co je statistický číselník?* [online].  
2014 [cit. 2014-05-12]. Dostupné z:  
[http://www.czso.cz/csu/klasifik.nsf/i/co\\_je\\_statisticky\\_  
ciselnik](http://www.czso.cz/csu/klasifik.nsf/i/co_je_statisticky_ciselnik)
- [25] GROFF, James a WEINBERG, Paul. *SQL Kompletní průvodce*  
Vyd. 1. Brno: CP Books, 2005, 936 s. ISBN 80-251-0369-2.
- [26] KANISOVÁ, Hana, MÜLLER, Miroslav. *UML srozumitelně*.  
2. aktualiz. vyd. Brno: Computer Press, 2006, 176 s. ISBN 80-251-1083-4.
- [27] ARLOW, Jim a NEUSTADT, Ila. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*  
Vyd. 1. Překlad Bogdan Kiszka. Brno: Computer Press, 2007, 567 s. ISBN 978-80-251-1503-9.
- [28] MATTEUCCI, Matteo. *Clustering – k-means* [online].  
2014 [cit. 2014-05-12]. Dostupné z:  
[http://home.deib.polimi.it/matteucc/Clustering/tutorial\\_html/  
kmeans.html](http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html)

## A Obsah přiloženého DVD

- Soubor Bakalarska\_prace\_2014\_Ludek\_Vesely.pdf
  - Text bakalářské práce
- Soubor Zadani\_bakalarske\_prace\_2014\_Ludek\_Vesely.pdf
  - Zadání bakalářské práce
- Složka client
  - Klientská aplikace
- Složka data
  - Soubor data.csv - stažená data ve formátu CSV
  - Soubor data.sql - skript pro vytvoření tabulek a vložení dat
- Složka datamodeler
  - Složka s logickým a relačním modelem v Data Modeleru
- Složka scripts
  - Složka se skripty, vstupními a výstupními soubory

## B Relační model databázového systému





## C Výsledek shlukové analýzy

